

Trabalho Prático 02 - AEDES 1 - Em Trios

Professora: Thais R. M. Braga Silva

Valor: 10 pontos

Data de Entrega: 30/10/2018

Forma de Entrega: PVANet (formato .zip ou .tar.gz)

O objetivo deste trabalho prático é permitir a avaliação do impacto causado pelo desempenho dos algoritmos em sua execução real. Vimos em sala de aula que existem problemas e algoritmos de complexidade exponencial, chamados de intratáveis. Nesses casos, os programas, ao serem executados podem demorar uma quantidade de tempo não razoável para encontrar uma solução, dependendo do tamanho da entrada. Vamos observar, portanto, como isso ocorre na prática. Para tanto, cada grupo fará uma implementação para o Problema do Caixeiro Viajante Assimétrico. Esse é um problema intratável, pois sua solução exata somente é possível através do cálculo e avaliação de todas as possíveis saídas, o que chamamos de força bruta. Em seguida, essa implementação deverá ser executada para diferentes tamanhos de entrada N , e o tempo gasto para que o programa termine, em cada caso, deverá ser medido por meio de comandos do sistema operacional.

Em linhas gerais, esse problema recebe como entrada N cidades e as distâncias entre todas elas. Considere que existe um caminho entre todos os pares de cidades, mas que o custo para ir da cidade i à cidade j não é o mesmo entre j e i . Com base nessas entradas, deve ser calculado qual o caminho com menor distância partindo de uma das cidades, percorrendo todas as demais, e retornando à cidade inicial. NOTA: as cidades são identificadas por números, que vão de 0 à $N-1$. Assim o conjunto de cidades é dado por $\{0,1,2,\dots,N-1\}$.

Existem diversas possíveis implementações para o problema do Caixeiro Viajante. Entretanto, as mais utilizadas requerem recursos de programação que ainda não foram estudados por vocês. Dessa forma, adotaremos uma estratégia mais direta, visto que o objetivo principal do trabalho é a avaliação de desempenho, e não o desenvolvimento do algoritmo. Cada dupla deverá implementar um programa, em linguagem C, para o problema do Caixeiro Viajante da seguinte forma:

- Procurar na Web um algoritmo de permutação para um conjunto de N elementos. Esse algoritmo deve, portanto, produzir todas as possíveis combinações entre os elementos N . Você deverá utilizar a linguagem C, portanto, se o algoritmo encontrado estiver em outra linguagem de programação, vocês deverão convertê-lo para linguagem C.
- Para armazenar as distâncias entre as N cidades do problema do Caixeiro Viajante, utilizar uma matriz M $N \times N$. Para a ligação entre

duas cidades i e j , adicionar os valores das distância entre elas nas posições $M[i][j]$ e $M[j][i]$. Lembre-se de que os valores $M[i][j]$ e $M[j][i]$ podem ser diferentes. As posições $M[i][i]$ da matriz devem receber 0. Exemplo: para $N = 3$, com $\text{distância}(0,1) = 5$, $\text{distância}(1,0) = 8$, $\text{distância}(0,2) = 10$, $\text{distância}(2,0) = 1$, $\text{distância}(1,2) = 2$ e $\text{distância}(2,1) = 4$, a matriz seria:

0	5	10
8	0	2
1	4	0

- Faça a soma de todos os dígitos dos números de matrícula dos integrantes do grupo. Em seguida faça o resto da divisão por N desse valor. O número X encontrado será usado como o ponto de partida e retorno. Dessa maneira, X não deverá ser considerado nas permutações a serem feitas, ou seja, o algoritmo de permutação será utilizado para $N-1$ valores, sendo que dentre estes, X não deve estar presente.
- Utilize o algoritmo de permutação para gerar as $(N-1)!$ possíveis combinações a partir do conjunto de $N-1$ cidades disponíveis.
- Para cada permutação resultante do passo anterior, utilize os valores da matriz M para calcular o custo do caminho. Considere a distância de X à primeira cidade e a distância entre a última cidade e X nesse cálculo. Escolha a permutação que possua o menor valor e apresente-a como solução exata do problema.
- Comece a executar o programa com o valor 5 para N . Em seguida, vá incrementando o valor de N de 1 em 1, registrando sempre a solução exata e o tempo gasto para encontrá-la. Ao executar o programa, utilize uma ferramenta para medição do tempo de execução, como o comando *time* do Unix.
- Utilize o arquivo *Exemplo.docx* disponibilizado como material auxiliar para este TP no PVANet para saber como deverá ser realizada a entrada e saída de dados. Caso sua implementação não esteja de acordo com as especificações passadas, haverá penalização na nota final do trio.
- Faça um relatório final contendo o código implementado, uma breve explicação do mesmo (indicando como funciona o algoritmo de permutação utilizado, de onde foi obtido, como o seu programa calcula os custos dos caminhos e escolhe o menor). O relatório também deverá conter os resultados dos tempos de execução para os valores de N testados pelo grupo, apresentados por meio de uma

tabela e de um gráfico (N x Tempo de Execução). Não se esqueça de colocar também no seu relatório as configurações da máquina em que os testes foram executados.

- Responda também no relatório à seguinte pergunta: Caso você fosse contratado por uma transportadora para desenvolver um sistema que encontrasse a menor rota a ser percorrida por seus caminhões, saindo e chegando do galpão de estoque e percorrendo uma única vez um conjunto de N localidades, você utilizaria a solução desenvolvida neste trabalho? Justifique a resposta.
- O arquivo do relatório final (pode ser um .doc, .pdf, .zip, .rar ou .tar.gz) deverá ser entregue até a data limite através do PVANet. Lembrem-se de colocar os nomes e matrículas dos dois integrantes da dupla.